

1 In the Claims:

2
3 1. (Currently Amended) A method of emulating memory management of
4 a low resource computing device with an emulator application running on a high
5 resource computing device, the method comprising:

6 ~~emulating an operation of a client;~~

7 allocating one or more contiguous portions of emulated memory on the
8 high resource computing device, wherein each contiguous portion has an address;

9 assigning an identifier to each address, wherein the identifier is assigned by
10 a memory manager associated with the emulator application;

11 entering each identifier into a table, wherein each identifier is mapped to its
12 corresponding address;

13 producing one or more pointers, wherein each of the one or more pointers
14 includes an identifier and points to an address corresponding to the identifier; and

15 emulating a memory access operation, wherein the emulating comprises:

16 attempting to access a contiguous portion of the emulated
17 memory having a first address;

18 comparing a first identifier in a pointer associated with the
19 first address ~~a pointer used by the emulated operation~~ with a second
20 identifier included in a ~~the table entry,~~ the second identifier being
21 mapped to the first address, wherein an address to a contiguous
22 portion of emulated memory is included in both the pointer and the
23 table entry; and

1 accessing the contiguous portion of emulated memory
2 associated with the first address ~~with the emulated operation~~ when
3 the first and second identifiers are the same.
4

5 2. (Currently Amended) The method as defined in Claim 1, wherein
6 the entering is performed by the memory manager associated with the emulator
7 application.

8 ~~the table entry is in a table that includes a plurality of said table entries;~~
9 ~~each said table entry references an address of one said contiguous portion~~
10 ~~of the emulated memory;~~

11 ~~the pointer is one of a plurality of said pointers; and~~

12 ~~each said pointer includes:~~

13 ~~the address of a respective said contiguous portion of the~~
14 ~~emulated memory; and~~

15 ~~one said identifier corresponding to the respective said~~
16 ~~contiguous portion of the emulated memory.~~
17

18 3. (Currently Amended) The method as defined in Claim 1, further
19 comprising outputting a diagnostic when the first and second identifiers are not the
20 same ~~wherein the accessing further comprises identically changing the identifier in~~
21 ~~both of the corresponding pointer to contiguous portion of emulated memory and~~
22 ~~the corresponding table entry when the permitted access is not a read or a write~~
23 ~~operation.~~
24
25

1 4. (Currently Amended) The method as defined in Claim 1, wherein
2 the ~~client~~ low resource computing device is selected from the group consisting of:
3 a personal computer (PC);
4 a workstation;
5 a server;
6 a set top box;
7 a video game console;
8 a Personal Digital Assistant (PDA);
9 a cellular telephone;
10 a handheld computing device; and
11 a computing device having less memory and/or computing resources than
12 that of another computing device executing ~~an~~ the emulator application ~~that~~
13 ~~emulates the operation of the client.~~

14
15 5. (Original) A computer-readable medium comprising instructions
16 that, when executed by a computer, performs the method of claim 1.
17

18 6. (Currently Amended) A method for emulating at least some of the
19 operational characteristics of a low resource computing device using resources of
20 a high resource computing device comprising:

21 making a call to a memory manager for an emulated memory access
22 operation to an allocated contiguous portion of emulated memory, wherein a
23 generation count has been assigned by the memory manager for the emulated
24 memory access operation to:
25

1 a plurality of table entries corresponding to a respective plurality of
2 said allocated contiguous portions of emulated memory, and
3 a plurality of pointers each including an address to a respective said
4 allocated contiguous portion of emulated memory;
5 comparing the generation count:
6 in the pointer including the address to the allocated contiguous
7 portion of emulated memory; and
8 in the table entry corresponding to the allocated contiguous portion
9 of emulated memory;
10 if the respective said generation counts in the comparison do not match,
11 then outputting a diagnostic; and
12 if the respective said generation counts in the comparison match, removing
13 the generation count from the pointer specified by the memory manager for the
14 emulated memory access operation during the performing of the emulated memory
15 access operation for which the memory manager was called.

16
17 7. (Original) The method as defined in Claim 6, further comprising:
18 performing the emulated memory access operation for which the memory
19 manager was called when there is a match of the respective said generation counts;
20 and
21 preventing the performance of the emulated memory access operation for
22 which the memory manager was called when the respective said generation counts
23 of the comparison do not match.
24
25

1 8. (Previously Presented) The method as defined in Claim 7, further
2 comprising, when there is a match and the emulated memory access operation is
3 not a read or a write operation, incrementing the generation count in both:

4 the pointer including the address to the allocated contiguous portion of
5 emulated memory; and

6 the table entry corresponding to the allocated contiguous portion of
7 emulated memory.

8
9 9. (Cancelled)

10
11 10. (Original) The method as defined in Claim 6, wherein the emulated
12 memory access operation is selected from the group consisting of:

13 a read operation;

14 a write operation;

15 a reallocation operation; and

16 an operation to free one or more of said allocated contiguous portions of
17 emulated memory.

18
19 11. (Previously Presented) The method as defined in Claim 6, further
20 comprising, prior to the making of the call:

21 making a call to the memory manager for to allocate a contiguous portion
22 of emulated memory;

23 receiving one said pointer from the memory manager that includes the
24 address of the allocated contiguous portion of emulated memory;

1 performing the allocation of the contiguous portion of emulated memory;
2 and
3 inserting the generation count:
4 in the:
5 the pointer including the address to the one said allocated
6 contiguous portion of emulated memory; and
7 the plurality of table entries corresponding to the one said
8 allocated contiguous portion of emulated memory.

9
10 12. (Cancelled).

11
12 13. (Original) A computer-readable medium comprising instructions
13 that, when executed by a computer, performs the method of Claim 12.

14
15 14. (Currently Amended) In a first computing device executing a first
16 application for the emulation of a second computing device executing a second
17 application, a method comprising:

18 making a call from the second application to a memory manager for an
19 emulated memory access operation to an allocated contiguous portion of emulated
20 memory used by the second application and including a plurality of said allocated
21 contiguous portions, wherein:

22 a generation count is in a plurality of table entries corresponding to a
23 respective plurality of said allocated contiguous portions of emulated
24 memory, the generation count being entered into the plurality of table
25 entries by the first application;

1 a generation count is in a plurality of pointers each including an
2 address to a respective said allocated contiguous portion of emulated
3 memory, the generation count being included in the plurality of pointers by
4 the first application;

5 for the emulated memory access operation, the memory manager
6 uses the address in the pointer that corresponds to the allocated contiguous
7 portion in emulated memory after removal of the generation count from the
8 pointer; and

9 prior to performing the emulated memory access operation to the allocated
10 contiguous portion of emulated memory:

11 comparing the generation count:

12 in the pointer including the address of the allocated
13 contiguous portion of the emulated memory; and

14 in the table entry corresponding to the allocated contiguous
15 portion of the emulated memory;

16 outputting a diagnostic when the respective said generation counts of
17 the comparison do not match.

18
19 15. (Original) The method as defined in Claim 14, further comprising:
20 performing the emulated memory access operation for which the memory
21 manager was called when there is a match of the respective said generation counts;
22 and

23 preventing the performance of the emulated memory access operation for
24 which the memory manager was called when the respective said generation counts
25 of the comparison do not match.

1
2 16. (Previously Presented) The method as defined in Claim 15 further
3 comprising, when there is a match of the respective said generation counts and the
4 emulated memory access operation is not a read operation or a write operation,
5 incrementing the generation count in both:

6 the pointer including the address to the allocated contiguous portion of
7 emulated memory; and

8 the table entry corresponding to the allocated contiguous portion of
9 emulated memory.
10

11 17. (Original) The method as defined in Claim 14, further comprising,
12 when:

13 the comparison finds that there is a match of the respective said generation
14 counts; and

15 the emulated memory access operation is neither a read operation nor a
16 write operation:

17 performing the emulated memory access operation for which the
18 memory manager was called and during which the generation count is
19 removed from the pointer used by the memory manager.
20

21 18. (Original) The method as defined in Claim 14, wherein the
22 emulated memory access operation is selected from the group consisting of:

23 a read operation;

24 a write operation;

25 a reallocation operation; and

1 an operation to free one or more of said allocated contiguous portions of
2 emulated memory.

3
4 19. (Previously Presented) The method as defined in Claim 14, further
5 comprising, prior to the making of the call by the second application to the
6 memory manager for the emulated memory access operation:

7 making a call by the second application to the memory manager for an
8 allocation of said allocated contiguous portion of emulated memory;

9 receiving one said pointer from the memory manager that includes an
10 address to said allocated contiguous portion of emulated memory;

11 performing an allocation of said allocated contiguous portions of emulated
12 memory; and

13 incrementing the generation count in both:

14 the pointer including the address to said allocated contiguous
15 portion of emulated memory; and

16 the table entry corresponding to said allocated contiguous
17 portion of emulated memory.

18
19 20. (Original) The method as defined in Claim 14, wherein the second
20 computing device is selected from the group consisting of:

21 a PC;

22 a workstation;

23 a server;

24 a set top box;

25 a video game console;

1 a PDA;
2 a cellular telephone;
3 a handheld computing device;
4 a consumer electronic device having a processor and memory; and
5 a client having less memory and/or computing resources than that of the
6 first computing device.

7
8 21. (Original) A computer-readable medium comprising instructions
9 that, when executed by a computer, performs the method of Claim 14.
10

11 22. (Currently Amended) A computer-readable medium including
12 instructions for execution by a computer, wherein the instructions comprise:
13 first logic calling for an emulated memory access operation with respect to
14 a first of a contiguous portion of an emulated memory for which there is:

15 a corresponding table entry in a table having a plurality of said table
16 entries that map to respective other said portions of the emulated memory,
17 wherein each said table entry includes an identifier, and further wherein
18 each identifier is entered into its corresponding table entry by an emulator
19 application; and

20 a corresponding pointer to a plurality of pointers each including an
21 identifier and an address to a respective said contiguous portion of the
22 emulated memory, wherein each identifier is entered into its corresponding
23 pointer by the emulator application;
24
25

1 second logic, in response to the first logic, such that, if the identifier in the
2 table entry corresponding to the first said contiguous portion is the same as the
3 identifier in the pointer corresponding to the first said portion, then:

4 the emulated memory access operation is performed with respect to
5 the first said contiguous portion of the emulated memory; ~~and~~

6 ~~when the emulated memory access operation is neither a read~~
7 ~~operation nor a write operation, the identifier is identically changed in both:~~

8 ~~the table entry corresponding to the first said portion; and~~

9 ~~the pointer corresponding to the first said portion;~~

10 ~~third logic, when the identifier in the table entry corresponding to the first~~
11 ~~said contiguous portion is different from the identifier in the pointer corresponding~~
12 ~~to the first said portion, calling for a diagnostic to be output.~~

13
14 23. (Original) The computer-readable medium as defined in Claim 22,
15 wherein the emulated memory access operation is selected from the group
16 consisting of:

17 a read operation;

18 a write operation;

19 a reallocation operation; and

20 an operation to free one or more of said portions of the emulated memory.

21
22 24. (Original) The computer-readable medium as defined in Claim 22,
23 wherein the performance of the memory operation further comprises removing the
24 identifier from the pointer corresponding to the first said contiguous portion
25 during the performance of the memory operation.

1
2 25. (Currently Amended) The computer-readable medium as defined in
3 Claim 22, wherein the second logic is further configured to identically change the
4 identifier in both the table entry corresponding to the first said portion and the
5 pointer corresponding to the first said portion when the emulated memory access
6 operation is neither a read operation nor a write operation. ~~A first apparatus to~~
7 ~~execute each said logic of Claim 22 so as to emulate a second apparatus executing~~
8 ~~an application using the emulated memory, wherein the second apparatus is~~
9 ~~selected from the group consisting of:~~

10 a PC;
11 a workstation;
12 a server;
13 a set top box;
14 a video game console;
15 a PDA;
16 a cellular telephone;
17 a handheld computing device; and
18 a client having less memory and/or computing resources than that of the
19 first apparatus.

20
21 26. (Currently Amended) A computer-readable storage medium having
22 computer-readable instructions stored thereon that, when executed, perform acts
23 comprising:
24 ~~first software program which, when executed by a computing device,~~
25 ~~emulates~~emulating the execution of a second application ~~software program using a~~

1 first application and emulated memory, the first application software program
2 comprising instructions that permit the second application software program to
3 perform an emulated memory access operation on a previously allocated
4 contiguous portion of the emulated memory only when a pointer and a table entry
5 both include ~~the~~ a same identifier entered by the first application, wherein:

6 the pointer also includes an address to ~~the~~ a previously allocated contiguous
7 portion which is useable to access the previously allocated contiguous portion
8 after removal of the identifier; and

9 the table entry maps to the previously allocated contiguous portion.

10
11 27. (Currently Amended) The computer-readable storage medium of
12 claim 26 ~~the first software program as defined in Claim 26~~, wherein:

13 the table entry is one of a plurality of said table entries that map to a
14 respective plurality of said portions of the emulated memory; and

15 the pointer is one of a plurality of said pointers that each include:

16 the address to a respective said contiguous portion of the emulated
17 memory; and

18 one said identifier entered by the first application corresponding to
19 the respective said contiguous portion of the emulated memory.

20
21 28. (Currently Amended) The computer-readable storage medium of
22 claim 26 further comprising instructions, that when executed allow the first
23 application to perform acts further comprising ~~first software program as defined~~
24 ~~in Claim 26, wherein the performance of the emulated memory access operation~~
25 ~~on the contiguous portion of the emulated memory further comprises:~~

1 removing the identifier from the corresponding pointer when it is processed
2 by the execution of the second application software program; and

3 when the emulated memory access operation is neither a read operation ~~not~~
4 nor a write operation, identically changing the identifier ~~with the first software~~
5 ~~program~~ in both of the corresponding pointer and table entry after the execution of
6 the second application software program has performed the emulated memory
7 access operation on the contiguous portion of the emulated memory.

8
9 29. (Currently Amended) The computer-readable storage medium of
10 claim 27 further comprising instructions, that when executed allow the first
11 application to perform acts further comprising first software program as defined in
12 ~~Claim 27, wherein the instructions further comprise~~ removing the identifier from
13 each said pointer prior to its use by the second application software program.

14
15 30. (Currently Amended) The computer-readable storage medium of
16 claim 27 further comprising instructions, that when executed perform acts further
17 comprising first software program as defined in Claim 27, wherein the
18 ~~instructions further comprise~~ allowing use of the table entries and identifiers with
19 the first application software program but not by the second application software
20 program.

21
22 31. (Cancelled)

23
24 32. (Currently Amended) A computer-readable medium including
25 instructions for execution by a computer, wherein the instructions comprise:

1 means for emulating an operation of a client as the client executes an
2 application; and

3 means for outputting a diagnostic when:

4 the emulated operation attempts to access a previously
5 allocated contiguous portion of emulated memory using a pointer including
6 an identifier placed in the pointer by the means for emulating, wherein the
7 pointer is configured to access the previously allocated contiguous portion
8 of the emulated memory upon removal of the identifier; and

9 a table entry used to manage the emulated memory does not
10 include the same identifier as the identifier in the pointer, wherein an
11 address to the previously allocated contiguous portion is included in both
12 the pointer and the table entry, and further wherein the identifier in the table
13 entry was placed in the pointer by the means for emulating.

14
15 33. (Previously Presented) The computer-readable medium as defined
16 in Claim 32, wherein:

17 the table entry is in a table that includes a plurality of said table entries;
18 each said table entry references an address of one said previously allocated
19 contiguous portion of the emulated memory;

20 the pointer is one of a plurality of said pointers; and

21 each said pointer includes:

22 the address to a respective said previously allocated
23 contiguous portion of the emulated memory; and

24 one said identifier corresponding to the respective said
25 previously allocated contiguous portion of the emulated memory.

1
2 34. (Currently Amended) The computer-readable medium as defined in
3 Claim 32, further comprising means for ~~permitted~~ permitting the attempted access
4 by the emulated operation to the previously allocated contiguous portion of
5 emulated memory, wherein during prior to said access:

6 the identifier is removed from the corresponding pointer to the contiguous
7 portion of emulated memory; and

8 when the permitted access is not a read or a write operation, the identifier in
9 both of the corresponding pointer to contiguous portion of emulated memory and
10 the corresponding table entry is identically changed.

11
12 35. (Previously Presented) The computer-readable medium as defined
13 in Claim 34, further comprising:

14 means, prior to an allocation of the previously allocated contiguous portion
15 of emulated memory, for making a call to a memory manager for an allocation of
16 the previously allocated contiguous portion of emulated memory;

17 means for receiving the pointer from the memory manager that includes the
18 address to the previously allocated contiguous portion of emulated memory;

19 means for performing the allocation of the previously allocated contiguous
20 portion of emulated memory;

21 means for inserting the generation count in the table entry; and

22 means for copying the generation count from the table entry to the pointer.

23
24 36. (Original) The computer-readable medium as defined in Claim 32,
25 wherein the client being emulated is selected from the group consisting of:

1 a PC;
2 a workstation;
3 a server;
4 a set top box;
5 a video game console;
6 a PDA;
7 a cellular telephone;
8 a handheld computing device; and
9 a computing device having less memory and/or computing resources than
10 that of another computing device executing an application that emulates the
11 operation of the client.
12
13
14
15
16
17
18
19
20
21
22
23
24
25